

SPECIFICATION

Prior to examination, please amend the application as follows:

Page 1 lines 1-2 delete and substitute per revised amendment practice using strikethrough and underline method the following:

~~This application claims the benefit of Provisional patent application serial number 60/263,436 filed January 23, 2001.~~

Cross reference to related application: This application is entitled to the benefit of PPA serial number 60/263,436 filed January 23, 2001.

Add the following three descriptions at the end of page 25.

CPU logic to store recently used data in transition buffer:

Prior art CPUs are designed with different logic concepts and circuits for storing recently used data in the cache memory. The new concept proposed in the present invention requires that the recently used data should be stored in the transition buffer. The CPU has counting device by which a certain number of instructions are counted. And as the program execution progresses, recently used data is updated. If the same instruction is implemented more than once, the instructions being repeated is not counted and next instruction is stored. At any given time the number of recently used data can be updated by software instructions or predefined arrangements made at the compile time. Thus a predefined number of data is available at any time for CPU to access to optimize performance and reduce power usage.

CPU logic to implement pipelined storage in main memory:

One of the unique features of the present invention is that logic is used in the CPU to implement different functions to optimize performance, reduce parts count and improve thermal management.

A pipelined storage is implemented in the main execution memory by the CPU using a special logic to implement multiway branches without execution delays. This logic gives

the CPU access to the instructions the CPU needs for implementing special branch instructions to execute the branch instructions without any execution delays.

At the compile time, deterministic requirements of the computer system is determined as to the sequence of proper instructions flow. The special logic to implement pipelines in the main execution memory determines at the correct sequence and proper time in the program, the preloading of instructions to implement multiway branch in a pipeline structure for the CPU to access.

When a particular branch or any other instructions are encountered by the CPU, all the different paths should be available to the CPU for proper execution in a deterministic mode. The special logic provides the instruction flow necessary to the CPU in a preloaded pipeline form offering multiple branch options. The CPU then determines the correct path of the branch and completes the execution of the instruction without any delay. This logic also has the capability of implementing pipeline storage in the transition buffer for the CPU to access. An additional logic is also implemented in the CPU to decode instructions located in pipeline storage in main execution memory or transition buffer in response to the CPU instructions. These instructions are decoded in this special logic in advance in response to decisions made during compile time or decision made during the execution of instructions in normal follow.

Storage area in CPU to store data related to main execution memory:

The CPU is provided with logic and storage area to store data that can be available to CPU at any time and can be related to main execution memory. This will allow the use of critical data by CPU more efficient without frequent memory accesses to the main execution memory. The logic and storage area to store data will selectively load the critical data from the main execution memory to the storage area in the CPU and selectively remove the data from the CPU. This can be determined at the execution or program and conditions determined at the compile time.

Add the following description, "conclusion and summary of advantages" on page 46 after line 6 and paragraph ending "...correction accordingly."

CONCLUSION AND SUMMARY OF ADVANTAGES

The primary objective of the present invention is to advance the state of the art of the computer architectures and technology by overcoming the limitations and disadvantages of prior art cache memory based computer systems.

This objective is achieved by proposing a novel architecture and circuit design concepts that eliminates use, function and purpose of cache memory.

From the concepts and data presented in the specification it is clear that the new architecture has provided definite improvements in many areas of architecture, design, performance, cost and power usage.

Higher performance is realized due to elimination of cache memory. Execution delays due to cache miss is eliminated. Further improvement in performance results in prevention of operating system crashes related to cache memories and data integrity problems. This gives the new architecture higher performance capabilities than prior art systems in terms of throughput and faster program execution. The deterministic operation allows prediction of performance bottlenecks in advance. Higher input/output capabilities and coherency is achieved. Data integrity is improved due to elimination of cache and computational errors are eliminated.

The overall system cost is also reduced due to use of low cost, but slower DRAM than currently used in prior art systems. The processor cost is also reduced due to capability of locating transition buffer outside the processor IC package. Cost per MIPS is further reduced due to synergistic effect of combining lower cost low power components.

Power usage is reduced since the transition buffer does not act like cache memory. Prior art cache memory program is executed from higher power consumption cache memory.

There is no transfer and duplication of instruction and data from main memory to the cache memory before execution takes place. This greatly reduces power consumption. Since the transition buffer need not be in “power on” mode like cache memory, the transition buffer can be put in quiescent mode more often. Power is further reduced since more program execution occurs from low power DRAM than high power transition buffer.

Definite performance, cost, power and size advantage is realized by using different types of memories in main execution memory and transition buffer. As described earlier in the specification, the access time of the transition buffer is closer to CPU instruction execution time. The access time of the DRAM used in main execution memory is multiple of CPU instruction execution time, depending on the number of DRAM is connected in parallel.

The low power, low cost of DRAM requires that the memory access time is greater than the SRAM transition buffer access time. By connecting DRAM in parallel, the access time of pre-accessed DRAM banks as described earlier in the specification is brought near the CPU instruction execution time and to manageable values. Thus the low power, low speed DRAM can be used at the same speed as the transition buffer SRAM.

The new architecture advances the state of the art both in areas of performance, power usage, reliability and cost.

The advantages of the new architecture can be summarized as follows:

NEW AND UNEXPECTED RESULTS PRODUCED BY THE NEW CACHELESS COMPUTER SYSTEM

1. Deterministic systems are always superior to probabilistic systems because they are more predictable and manageable and are easy to design.

2. CPU can execute programs at CPU speed without interruption since there is no need for cache memory to supply required instruction flow and data.
3. Eliminates the need for on chip cache memory and related control and management logic needed to manage on chip cache memory, making dramatic reduction in the total wafer size and heat dissipation.
4. Eliminates the need for on chip cache memory making dramatic improvements in the total power usage, heat dissipation and resulting thermal management.
5. Total die size for the processor is greatly reduced due to improvements in processor pipelining schemes and control logic.
6. The speed and performance is not related to size and speed of the cache memory.
7. Real time operation is possible for all the systems including the DSP.
8. Real time “on the fly” program transfer to the memory and instant execution of the program due to the availability of details of program execution sequence in advance.
9. Data integrity and consistency problems are eliminated because there is no need to keep two copies of data in cache memory as well as main semiconductor memory, creating conflicts between the stale data and recently updated data.
10. Server architectures with multiple processors have improved performance due to elimination of problems related to inter-processor communication. In addition, problems relating to data integrity and consistency are also eliminated.
11. Greatly improves fault tolerance and reliability for single processor and multiprocessor systems.
12. The size of the programs that can be executed without interruption and conflicts in data integrity is equal to the size of the entire main semiconductor memory and not just the size of the cache memory.
13. Pipeline restart and branch prediction is deterministic which avoids execution delays and reduces heat dissipation and power usage
14. FPU pipeline restart is deterministic which avoids execution delays and reduces heat dissipation and power usage.

15. FPU pipeline depth can be optimized for each FPU operation in advance due to deterministic information available, which avoids execution delays and reduces heat dissipation and power usage.
16. The power management is brought in to the deterministic domain. This means the power usage and thermal behavior can be predicted at the compile time.
17. Power is further reduced since more program execution occurs from low power DRAM than high power transition buffer.
18. The power usage and thermal behavior envelope can be predicted at the compile time and queries pertaining to that behavior can be made at the compile time and necessary adjustments can be made in advance.
19. The CPU speed can be reduced at the exact point in the program execution due to prior information obtained at the compile time to improve power usage and thermal management.
20. Existing software can be used without modification with greater performance capabilities than obtained previously.
21. Performance bottlenecks can be predicted at the compile time and necessary adjustments can be made in advance, thereby avoiding timing problems and system crashes.